

Суть алгоритма укладки звездой состоит в следующем. Помещаем все единицы в одну плотноупакованную гидрофобную (стабильную) структуру. Основа такой структуры составлена не более чем из шести единиц на каждом уровне. Такая структура составлена из уровней, каждый из которых представляет фрагмент двумерной триангулированной решетки. Показано, что алгоритм укладки звездой имеет оценку приближения, равную 1.875.

Графом квадратной двумерной решетки с диагоналями назовем граф $G = (V(G), E(G))$, для которого выполняются следующие условия:

$$V(G) = Z^2 = Z \times Z,$$

$$E(G) = \{ \{u, v\} \mid u, v \in V(G), \|u - v\|_2 \leq \sqrt{2} \}.$$

Иллюстрация работы алгоритма остовной укладки для двумерной квадратной решетки с диагоналями приведена на рис. 3. Доказано, что остовный алгоритм для данной решетки является 3-приближенным.

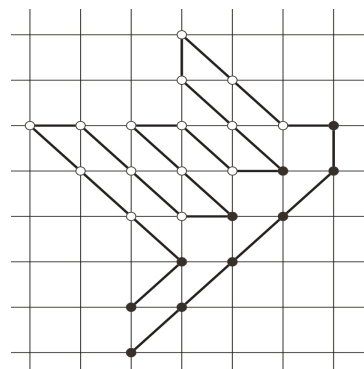


Рис. 3. Пример остовной укладки для двумерной квадратной решетки с диагоналями

Литература

1. *Branden C.-I., Tooze J.* Introduction to protein structure / Garland Publishing. 1999.
2. *Baxevanis A., Francis Ouellette B.F.* Bioinformatics: a practical guide to the analysis of genes and proteins / John Wiley & Sons, Inc. 1998.
3. *Rashidi H.H., Buehler L.K.* Bioinformatics basics applications in biological science and medicine / CRC Press. 1999.
4. *Agarwala R., Batzoglou S., Dan V., Decatur S.E., Farach M., Hannenhalli S., Skiena S.* Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model / J. Comput. Biology. 1997. Vol. 4. P. 275–296.
5. *Dill K.A.* Theory for the folding and stability of globular proteins / Biochemistry. 1985. Vol. 24. P. 1501.
6. *Crescenzi P., Goldman D., Papadimitriou C., Piccolboni A., Yannakakis M.* On the complexity of protein folding / J. Comput. Biology. 1998. Vol. 5. P. 423–466.

РАСПАРАЛЛЕЛИВАНИЕ АЛГОРИТМА ЧИСЛЕННОГО РЕШЕНИЯ УРАВНЕНИЯ ТЕПЛОПРОВОДНОСТИ ПЕРВОГО ПОРЯДКА

А. Ю. Лобанов

В связи с тем, что физические возможности совершенствования вычислительных устройств ограничены, в последнее время многие надежды на увеличение производительности связаны со специализированными

процессорами, обеспечивающими высокую степень соответствия алгоритма и архитектуры.

Однако процесс изготовления спецпроцессоров на основе СБИС накладывает определенные ограничения на их архитектуру. В связи с чем, встают задачи адаптация существующих алгоритмов для реализации на СБИС и разработка новых алгоритмов, приспособленных для реализации на СБИС.

В данной работе рассматривается проектирование линейного спецпроцессора для алгоритма численного решения уравнения теплопроводности первого рода.

Смешанная задача первого рода [1] для уравнения теплопроводности выглядит следующим образом:

$$\frac{\partial u(x,t)}{\partial x} = a \frac{\partial^2 u}{\partial x^2} + f(x,t), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq T, \quad (1)$$

$$u(x,0) = f^{(0)}(x), \quad 0 \leq x \leq 1, \quad (2)$$

$$\begin{aligned} u(0,t) &= \mu_0(t), \quad 0 \leq t \leq T, \\ u(1,t) &= \mu_1(t), \quad 0 \leq t \leq T \end{aligned} \quad (3)$$

Для решения задачи (1-3) используется следующая двухслойная явная разностная схема [1]:

$$\begin{aligned} y(i,j) &= \alpha y(i-1,j-1) + \beta y(i,j-1) + \alpha y(i+1,j-1), \\ 1 \leq i \leq N-1, \quad 1 \leq j \leq M, \\ y(0,j) &= \mu_0(j\tau), \quad 1 \leq j \leq M, \\ y(N,j) &= \mu_1(j\tau), \quad 1 \leq j \leq M, \\ y(i,0) &= f^{(0)}(ih), \quad 0 \leq i \leq N \end{aligned} \quad (4)$$

где $\alpha = \frac{a\tau}{h^2}$, $\beta = 1 - \frac{a\tau}{h^2}$. Полученная схема устойчива при $\alpha = \frac{a\tau}{h^2} < \frac{1}{2}$.

Однако данный алгоритм не является схемой с однократным присваиванием [2]. Привести его к схеме с однократным присваиванием можно путем увеличения числа переменных:

$$\begin{aligned}
y^{(k)}(i, j) &= \alpha y^{(1)}(i-1, j-1) + \beta y^{(2)}(i, j-1) + \alpha y^{(3)}(i+1, j-1), \\
1 \leq i \leq N-1, 1 \leq j \leq M, 1 \leq k \leq 3, \\
y^{(k)}(0, j) &= \mu_0(j\tau), \quad 1 \leq j \leq M, 1 \leq k \leq 3, \\
y^{(k)}(N, j) &= \mu_1(j\tau), \quad 1 \leq j \leq M, 1 \leq k \leq 3, \\
y^{(k)}(i, 0) &= f^{(0)}(ih), \quad 0 \leq i \leq N, 1 \leq k \leq 3
\end{aligned} \tag{5}$$

Множество векторов зачисимости имеет вид:

$$\Phi = \{\varphi_1 = (1,1), \varphi_2 = (0,1), \varphi_3 = (-1,1)\}, \tag{6}$$

Построим граф зависимостей [2] для алгоритма (4) $G = (V, E)$:

$$V = \{(v_1, v_2) \in Z^2, 1 \leq v_1 \leq N-1, 1 \leq v_2 \leq M\}, \tag{7}$$

$$E = \{(v_1, v_2) \in V \times V, v_2 = v_1 + \varphi, \varphi \in \Phi\}, \tag{8}$$

Граф зависимостей $G = (V, E)$ является строго направленным [2], так как конус допустимых направлений

$$K(G) = \{\tau = (\tau_1, \tau_2) \in Z^d \mid \tau \cdot \varphi \geq 1, \varphi \in \Phi\}, \tag{9}$$

не пуст. Принадлежность вектора конусу допустимых направлений определяется системой неравенств:

$$\begin{cases} \tau_2 - \tau_1 \geq 1, \\ \tau_2 \geq 1, \\ \tau_1 + \tau_2 \geq 1 \end{cases}, \tag{10}$$

Для получения линейного спецпроцессора, зададим отображение $\pi : Z^2 \rightarrow Z^1$, (11)

Функция (11) называется функцией размещения. Согласно требованиям СБИС архитектуры [2], обмен данными может выполняться только между соседними в линейке процессорными элементами, что можно записать в виде:

$$\pi\varphi \in \{0, \pm 1\}, \varphi \in \Phi, \tag{12}$$

Введя дополнительное условие на неотрицательность (11), получим два возможных варианта функции размещения:

$$\pi = (0 \ 1), \tag{13}$$

$$\pi = (1 \ 0), \tag{14}$$

В данной работе рассматривается случай (13).

Для корректного ввода граничных условий введем две дополнительные переменные, транспортируемые по направлению φ_3 , Область вычисления при этом расширяется в положительном направлении оси v_1 .

$$\mu_0(v) = \mu_0(v - \varphi_3), \quad \mu_1(v) = \mu_1(v - \varphi_3), \quad (15)$$

$$v \in \bar{V} = \{(v_1, v_2) \in Z^2, 1 \leq v_1 \leq N + M - v_2 - 1, 1 \leq v_2 \leq M\}, \quad (16)$$

Для организации управления используется механизм управляющих меток, которые доставляются в вершины по принципу магистральной рассылки [2].

В нашем случае (при условии $N > 2$), присутствует три вида операций, которые должен производить каждый процессорный элемент:

- вычисление, с использованием левого граничного условия
- вычисление, без использования граничных условий
- вычисление, с использованием правого граничного условия

Для этого, введем дополнительную переменную управления, которая будет передавать управляющие метки m_1 , m_2 и m_3 , соответственно. Метки будут передаваться по направлению φ_2 .

В первый такт на соответствующий вход первого процессора подается метка m_1 . Далее, до такта $N-2$ подаются метки m_2 , а в такт $N-1$ подается метка m_3 .

Таймирующая функция [2] для параллельной формы алгоритма будет выбираться из требования минимизации высоты параллельной формы

$$\begin{aligned} T(\tau) &= \max_{v \in \bar{V}} t(v) - \min_{v \in \bar{V}} t(v) + 1 = \max_{s \in \bar{V} - \bar{V}} \tau \cdot s + 1, \\ \bar{V} - \bar{V} &= \left\{ s \in Z^2 \mid s = v' - v'', v', v'' \in \bar{V} \right\}, \quad \tau \in K(G) \end{aligned} \quad (17)$$

и условия совмести таймирующей функции и функции размещения π :

$$\tau \cdot s \neq 0, \tau \in K(G), \text{ если } \pi(s) = 0, s \in \bar{V} - \bar{V}, \quad (18)$$

что при $\pi = (0 \ 1)$ равносильно $\tau_1 \neq 0$.

Исходя из этого, получаем следующую задачу на минимум:

$$\begin{cases} T(\tau) = \max_{v \in \bar{V} - \bar{V}} \tau \cdot v + \max_k \tau \cdot \varphi_k \rightarrow \min, \\ 1 - \tau_2 \leq \tau_1 \leq \tau_2 - 1, \\ \tau_2 \geq 1, \\ \tau_1 \neq 0 \end{cases}, \quad (19)$$

Решив (19), получаем, что при $\tau = (1, 2)$, достигается минимальная высота параллельной формы алгоритма.

Для описания функционирования локальной памяти, необходимо ввести задержки по направлениям [2], определяющие количество тактов, на которое задерживается вычисленное значение перед передачей по данному направлению.

Задержки по направлениям будут равны:

$$\begin{aligned} h_{\varphi_1}(v) &= (1, 2) \cdot (1, 1) = 3, \\ h_{\varphi_2}(v) &= (1, 2) \cdot (0, 1) = 2, \\ h_{\varphi_3}(v) &= (1, 2) \cdot (-1, 1) = 1 \end{aligned} \quad (20)$$

Время реализации алгоритма будет равно:

$$T_{\min} = (1, 2) \cdot ((N - 2), (M - 1)) + 3 = N + 2M - 1, \quad (21)$$

Функционирование локальной памяти процессорного элемента описывается следующим образом:

$$\begin{aligned} R_1^{y(k)}(t) &= \alpha(I_{\varphi_1}^{\mu_0} + I_{\varphi_3}^{y(3)}) + \beta I_{\varphi_2}^{y(2)}, \quad R_1^m(t) = m_1, \quad 1 \leq k \leq 3, \\ R_1^{y(k)}(t) &= \alpha(I_{\varphi_1}^{y(1)} + I_{\varphi_3}^{y(3)}) + \beta I_{\varphi_2}^{y(2)}, \quad R_1^m(t) = m_2, \\ R_1^{y(k)}(t) &= \alpha(I_{\varphi_1}^{y(1)} + I_{\varphi_2}^{\mu_1}) + \beta I_{\varphi_2}^{y(2)}, \quad R_1^m(t) = m_3, \\ O_{\varphi_k}^{y(k)} &= R_{4-k}^{y(k)}(t-1), \quad R_2^{y(1)}(t) = R_1^{y(1)}(t-1), \quad R_1^{\mu_0}(t) = I_{\varphi_3}^{\mu_0}, \\ O_{\varphi_3}^{\mu_0} &= R_1^{\mu_0}(t-1), \quad R_2^{y(2)}(t) = R_1^{y(2)}(t-1), \quad R_1^{\mu_1}(t) = I_{\varphi_3}^{\mu_1}, \\ O_{\varphi_3}^{\mu_1} &= R_1^{\mu_1}(t-1), \quad R_3^{y(1)}(t) = R_2^{y(1)}(t-1), \quad R_1^m(t) = I_{\varphi_2}^m, \\ O_{\varphi_2}^m &= R_2^m(t-1), \quad R_2^m(t) = R_1^m(t-1), \end{aligned}$$

Таким образом, в данной работе построен параллельный алгоритм численного решения одномерного уравнения теплопроводности, и спроектирован спецпроцессор на СБИС для реализации этого параллельного алгоритма. Спецпроцессор состоит из M процессорных элементов, общее время работы алгоритма $T_{\min} = N + 2M - 1$.

Литература

1. Вакульчик П.А. Методы численного анализа: Курс лекций / Мн.: БГУ, 2002. 268с.
2. Кун С. Матричные процессоры на СБИС: Пер. с англ. / М.: Мир, 1991. 672с.